Extracting Template for Knowledge-based Question-Answering Using Conditional Random Fields

Changki Lee, Ji-Hyun Wang, Hyeon-Jin Kim, Myung-Gil Jang

Knowledge Mining Research Team
Speech/Language Technology Research Department
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, Korea
{leeck, jhwang, jini, mgjang}@etri.re.kr

Abstract. In this paper, we present an information extraction system that extracts template elements for a question-answering (QA) system in the domain of encyclopedia. We use Conditional Random Fields to extract templates from the texts of an encyclopedia. Using the proposed approach, we could achieve a 74.89% precision and a 55.77% F1 in the template extraction. In the question classification, we could archive an 83.6% precision and a 65.4% recall. Finally, in the Knowledge-based QA (including template extraction procedure), we could archive an 81.3% precision and a 33.3% recall. The result demonstrated that our approach is feasible and effective for template extraction for QA.

1. Introduction

The Question-Answering (QA) system is an information retrieval system that finds an answer instead of finding a list of documents in response to a user's question. The current QA approaches can be classified into two groups; Information Extraction (IE)-supported QA [3] known as template and the passage extraction methods based on passage retrieval system [2]. In the open-domain QA systems, the method of IE-supported QA is an impractical solution due to the dependency of IE systems on domain knowledge. However, in a closed domain such as encyclopedia texts, the combination of IR and IE system is successful [1, 2]. Passage extraction methods have been the most commonly used by many QA systems. In the passage extraction methods, sentences or passages that are regarded as the most relevant sentences or passages to the question are extracted and then answers are retrieved by using lexico-syntactic information or NLP techniques. However, it takes a long time to extract an answer in these QA systems because rules should be applied to each sentence including answer candidates on the retrieval time [5].

In this paper, we propose a template extraction system for a Knowledge-based QA system based on the encyclopedia Knowledge Base (KB) using Conditional Random Fields. We structure the remaining part of the paper as follows. In section 2, we describe the Conditional Random Fields, and in section 3, we describe our template extraction methodology, respectively section 4 presents the question analysis for

knowledge-based QA. Section 5 presents the experiments and experimental results. Section 6 concludes the research.

2. Conditional Random Fields

Conditional Random Fields (CRFs) [4] are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes.

In the special case in which the designated output nodes of the graphical model are linked by edge in a linear chain, CRFs make a first-order Markov independence assumption among output nodes, and thus correspond to finite state machines (FSMs).

Let $\mathbf{o} = <o_1, o_2, \ldots o_T>$ be some observed input data sequences, such as a sequence of words of sentences in a document. Let $\boldsymbol{S}$ be a set of FSM states, each of which is associated with a label, $l \in L$, (such as a label "BP (birthplace)" in Figure 1). Let $\mathbf{s} = <s_1,s_2,\ldots s_T>$ be some sequences of states, (the values on T output nodes). CRFs define the conditional probability of a state sequence given an input sequence as

$$P_\Lambda(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z_o} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t) \right), \quad (\mathbf{1})$$

where Z_o is a normalization factor over all state sequences, $f_k(s_{t-1},s_t,\mathbf{o},t)$ is an arbitrary feature function over its arguments, and λ_k is a learned weight for each feature function. A feature function may be defined to have a value 0 in most cases, and have a value 1 if and only if s_{t-1} is in state#1 (that may have label "O (outside)") and the observation at position $t-2$ in o is a word "graduated". Higher λ weights make their corresponding FSM translations more likely.

Z_o is the sum of the scores of all possible state sequences and is defined as follows:

$$Z_o = \sum_{s \in S^T} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t) \right). \quad (\mathbf{2})$$

The number of state sequences grows up exponentially as the length of input sequences, T, increases. In linear-chain-structured CRFs, as in forward-backward for HMMs, the probability that a particular transition was taken between two CRF states at a particular position in the input sequence can be calculated efficiently by dynamic programming [4].

2.1 Training CRFs

The {λ…} weights of a CRF are set to maximize conditional log-likelihood of labeled sequences in some training set, $D=\{<\mathbf{o},\mathbf{l}>^{(1)},\ldots,<\mathbf{o},\mathbf{l}>^{(N)}\}$:

$$L_\Lambda = \sum_{j=1}^{N} \log\left(P_\Lambda(\mathbf{l}^{(j)} \mid \mathbf{o}^{(j)})\right) - \sum_k \frac{\lambda_k^2}{2\sigma^2} \quad (3)$$

where the second sum is a Gaussian prior over parameters that provides smoothing to help cope with sparsity in the training data [4].

It has been shown that quasi-Newton methods, such as L-BFGS, are significantly more efficient than tradition iterative scaling and even conjugate gradient [7]. This method approximates the second-derivative of the likelihood by keeping a running, finite-sized window of previous first-derivatives.

L-BFGS can simply be treated as a black-box optimization procedure, requiring only that one provide the first-derivative of the function to be optimized. Assuming that the training labels on instance *j* make its state path unambiguous, let $s^{(j)}$ denote that path, and then the first-derivative of the log-likelihood is

$$\frac{\delta L}{\delta \lambda_k} = \left(\sum_{j=1}^{N} C_k(\mathbf{s}^{(j)}, \mathbf{o}^{(j)})\right) - \left(\sum_{j=1}^{N} \sum_s P_\Lambda(\mathbf{s} \mid \mathbf{o}^{(j)}) C_k(\mathbf{s}, \mathbf{o}^{(j)})\right) - \frac{\lambda_k}{\sigma^2} \quad (4)$$

where $C_k(\mathbf{s},\mathbf{o})$ is the count for feature *k* given **s** and **o**, equal to the sum of $f_k(s_{t-1}, s_t, \mathbf{o}, \mathbf{t})$ values for all positions, *t*, in the sequence **s**. The first two terms correspond to the difference between the empirical expected value of feature f_k and the model's expected value.

The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of independence assumptions required by HMMs. Additionally, CRFs avoid the label bias problem, a weakness exhibited by maximum entropy Markov models (MEMMs). CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks [4, 6].

3. Extracting Templates with CRFs

The purpose of this step is to build an encyclopedia Knowledge Base (KB). To construct the encyclopedia KB, templates for encyclopedia categories are required. For example, templates of the person category are person, career, scholarship, debut, and championship. We have defined 58 templates and 221 template elements for 19 categories. Table 1 shows templates and template elements for the person category.

Table 1. Template for person category.

Template	Template elements
Person	birth date, birth place, death date, death place, death reason, family origin, main achievement, activity field
Career	organization, joining date, position

Scholarship	graduate date, graduate school
Debut	debut journal, debut date, debut work
Championship	Championship name, Championship date

In this paper, we do not use a feature selection method and all features are used in training and testing phases. We use the following factored representation for features

$$f(s_{t-1}, s_t, \mathbf{o}, t) = p(\mathbf{o}, t)q(s_{t-1}, s_t) \quad \textbf{(5)}$$

where p(**o**,t) is a predicate on the input sequence **o** and current position *t* and $q(s_{t-1},s_t)$ is a predicate on pairs of labels.

Table 2. Tempalte extraction features

$q(s_{t-1},s_t)$	p(**o**,t)
s_t = s s_t = s & s_{t-1} = s'	true
s_t = s	$word_t$ = word $word_{t-1}$ = word $word_{t+1}$ = word $word_{t-2}$ = word $word_{t+2}$ = word tag_t = tag tag_{t-1} = tag tag_{t+1} = tag tag_{t-2} = tag tag_{t+2} = tag tag_{t-2} = tag & tag_{t-1} = tag' tag_{t-1} = tag & tag_t = tag' tag_t = tag & tag_{t+1} = tag' tag_{t+1} = tag & tag_{t+2} = tag' $suffix1_t$=suffix1(w) (where | suffix1(w) | = 1) $suffix2_t$=suffix2(w) (where | suffix2(w) | = 2) $suffix3_t$=suffix3(w) (where | suffix3(w) | = 3) $chunk_{t-1}$=chunk & $chunk_t$=chunk' $chunk_t$=chunk & $chunk_{t+1}$=chunk' last_verb=verb (where v is the last verb of a sentence)

Table 2 summarizes a feature set. For a given position *t*, $word_t$ is the word, tag_t is POS tag, s_t is label, $chunk_t$ is chunk label, $suffix_t$ is suffix, and last_verb is the last verb of a given sentence. For example, we extract features from a given sentence 'He $_1$ was $_2$ born $_3$ in $_4$ **Seonsan** $_5$, $_6$ Kyungbuk $_7$ Province $_8$' and given position 5 as follows:

word-2=born word-1=in word=Seonsan word+1=, word+2=Kyungbuk tag-2=VBD tag-1=IN tag=NNP tag+1=, tag+2=NNP tag-2_tag-1=VBD_IN tag-1_tag=IN_NNP tag_tag+1=NNP_, tag+1_tag+2=,_NNP suffix1=n suffix2=an suffix3=san chunk-1_chunk=O_B chunk_chunk+1=B_I verb=born

<Encyclopedia>
Title: Park Chung-Hee
Text: He was born in Seonsan, Kyungbuk Province.
Text: In 1937, he graduated at Taegu Normal school.

↓ Generating features for a given sentence.

Text: He was born in **Seonsan**, Kyungbuk Province.
Feature: word-2=born word-1=in word=Seonsan word+1=, word+2=Kyungbuk tag-2=VBD tag-1=IN tag=NNP tag+1=, tag+2=NNP … verb=born

↓ Labeling sentences using CRF.

Text (o_1): He was born in Seonsan , Kyungbuk Province .
Label (s_1): O O O O BP BP BP BP O
Text (o_2): In 1937 , he graduated at Taegu Normal school .
Label (s_2): O GD O O O O GS GS GS O
$P(s_1|o_1)=0.72$, $P(s_2|o_2)=0.89$

↓ Extracting template.

<Knowledge Base>

Title	**Template - Element**	**Record**	**Score**
Park Chung-Hee	Person – birth place (BP)	Seonsan, Kyungbuk Province	0.72
Park Chung-Hee	Scholarship - graduation school (GS)	Taegu Normal School	0.89
Park Chung-Hee	Scholarship - graduation date (GD)	1937	0.89

Fig. 1. The process of extracting template.

Figure 1 shows the process of extracting templates. First, we generate features from the words in a sentence of encyclopedia. And then, the words in a sentence are

tagged with their corresponding labels (ex, BP, GS, GD, O, etc) using CRF. Last, we extract template elements from the labels, and save them into Knowledge Base. We select the template element that the score is high in the case that the template element is duplicated. We use the probability of a sentence that includes the template element as the score (i.e. score = P(s|o)).

4. Question Analysis for Knowledge-based QA

The function of this step is to extract an answer matched with the clues in the question from encyclopedia KB. Figure 2 shows a couple of examples for answer processing.

1. **Question:** Where was Park Chung-Hee born?
   Entity: Person - birth place (BP)
   Entry: Park Chung-Hee
   Answer: Seonsan, Kyungbuk Province (score=0.72)

2. **Question:** What university did Park Chung-Hee graduate?
   Entity: Scholarship - graduation school (GS)
   Entry: Park Chung-Hee
   Answer: Taegu Normal School (score=0.89)

Fig. 2. Answer Processing.

Given a question sentence, we first parse the question, and then use the following rules to determine the category (i.e. template element) of the desired answer and to extract a main title of encyclopedia text:

- Where was X born → **Template-element** = Person **-** Birth place, **Title** = X.
- When was X born → **Template-element** = Person **-** Birth date, **Title** = X.
- What university did X graduate → **Template-element** = Scholarship - Graduation school, **Title** = X.
- When did X graduate → **Template-element** = Scholarship - Graduation date, **Title** = X.
- When did X die → **Template-element** = Person **-** Death date, **Title** = X.
- Where did X die → **Template-element** = Person **-** Death place, **Title** = X.
- … (1273 lexico-syntactic rules).

We then search a set of pairs (title = Park Chung-Hee, template-element = birthplace or graduation school) from the encyclopedia KB. We rank the extracted answers by their extraction scores, because there are more than one answer or there are conflicting answers.

5. Experiments

The experiments for extracting templates were performed on our Korean encyclopedia data set. The data set consists of 5,000 documents tagged by human annotators. We used 4,500 documents as a training set and 500 documents as a test set, respectively. We trained the model by L-BFGS using our C++ implementation of CRF. We use a Gaussian prior of 10 and 60 iterations.

Table 3. Performance of Extracting Template using CRFs.

Category	Precision	Recall	F1
Animals and plants	78.89	71.70	75.13
Buildings	73.33	50.00	59.46
Books	84.75	52.63	64.94
History	75.43	39.52	51.87
Organ	53.21	27.88	36.59
Overall	74.89	44.42	55.77
Overall (MaxEnt)	71.74	39.07	50.59

Table 3 shows the performance of extracting templates using CRFs. In "Animals and plants" category, we obtained the best performance, 75.13% F1. In the overall data, we could achieve a 74.89% precision, a 44.42% recall, and a 55.77% F1. We compared CRFs with an alternative model, Maximum Entropy (MaxEnt). In the case of Maxent, we archived a 71.74% precision, a 39.07% recall, and a 50.59% F1. From this comparison, we can see that CRFs outperforms Maxent.

Many errors occur in the long template elements. For example, 'industry product' has 'potato, tobacco, sugar cane, rape, and tea'. Moreover, long template elements are labeled inconsistently in the training data. Other errors could be fixed with additional feature engineering – for example, including additional specialized regular expressions, such as '[0-9][0-9][0-9]*', can make 'birth date' accuracy better. And increasing the amount of training data would be expected to help significantly. In most cases, the precision has better performance than recall. So, we plan to research to improve recall in the future.

To experiment on question analysis system, we use 78 questions of ETRI QA Test Set which consists of 200 pairs of question and answer. ETRI QA Test Set was created from the Doosan encyclopedia text collections, which is the most famous encyclopedia in Korea. In the question classification, we could archive an 83.6% precision and a 65.4% recall. Finally, in the Knowledge-base QA (including template extraction procedure), we could archive an 81.3% precision and a 33.3% recall.

Many errors occur in the question classification. Our question classification system is largely hand-built. The system is consisted of 1,273 lexico-syntactic rules. It is likely that a trainable machine learning approach would be a more feasible solution to

this problem. So, we plan to develop a trainable question analysis system using MaxEnt or CRF.

6. Conclusion

In this paper, we proposed a novel method to extract templates for encyclopedia QA. We used CRFs to extract templates from the text of encyclopedia. Using the proposed approach, we could achieve a 74.89% precision, a 44.42% recall, and a 55.77% F1 in the template extraction. In the question classification, we could archive an 83.6% precision and a 65.4% recall. Finally, in the Knowledge-based QA (including template extraction procedure), we could archive an 81.3% precision and a 33.3% recall. The result demonstrated that our approach is feasible and effective for template extraction for QA.

7. Reference

1. Julian Kupiec. *MURAX: A Robust Linguistic Approach for Question Answering Using On-line Encyclopedia*, SIGIR, 1993.
2. Sanda M. Harabagiu, Steven J. Maiorano. *Finding Answers in Large Collections of Texts: Paragraph Indexing + Abductive Inference*, AAAI, 1999.
3. Wei Li, Rohini K. Srihari. *Extracting Exact Answers to Questions Based Structural Links*, Coling, 2002.
4. J. Lafferty, A. McCallum, and F. Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML, 2001.
5. Harksoo Kim, Jungyun Seo, *A Reliable Indexing Method for a Practical QA System*, Coling, 2002.
6. David Pinto, A. McCallum, Xing Wei, and W. Bruce Croft. *Table Extraction Using Conditional Random Fields*. SIGIR, 2003.
7. S. Fei, F. Pereira. *Shallow Parsing with Conditional Random Fields*, HLT & NAACL, 2003.